

1 REMARKS

2 Status of the Claims

3 Claims 21-50 remain pending in the application, Claims 1-20 having been previously canceled.
4 Claim 40 has been amended to correct a minor typo error.

5 Claims Rejected under 35 U.S.C. § 103(a)

6 The Examiner has rejected Claims 21-50 as being anticipated by Doyle et al. (U.S. Patent
7 No. 5,838,906, hereinafter referred to as “Doyle”) in view of Held et al. (U.S. Patent No. 5,802,367,
8 hereinafter referred to as “Held”). The Examiner asserts that both Doyle and Held teach systems allowing one
9 program to call upon another program to perform an operation on a piece of data. The Examiner notes that
10 Doyle displays a list of applications and enables an application to be selected to run with an original program.
11 The Examiner further indicates that the selected application can enable an output from an input that the
12 original program would be unable to provide. He admits that Doyle does not expressly disclose a service
13 manager, but he asserts that a service manager “must exist” and further relies upon Held’s disclosure of a
14 “service control manager,” which the Examiner asserts is equivalent to the service manager. The Examiner
15 further asserts that Held’s surrogate process is equivalent to applicants’ recited service container. Applicants
16 respectfully disagree for the following reasons. The Examiner concludes that it would have been obvious to
17 one skilled in the art, at the time of the invention, to have combined the teachings of Doyle with those of Held,
18 to provide a method for transparently executing code using a surrogate process (Held, column 3, line 67 –
19 column 4, line 1). Applicants respectfully disagree with the rejection for the reasons explained below.

20 In the interest of reducing the complexity of the issues for the Examiner to consider in this response,
21 the following discussion focuses on independent Claims 21, 30, 36, and 50. The patentability of each
22 remaining dependent claim is not addressed in detail; however, applicants’ decision not to discuss the
23 differences between the cited art and each dependent claim should not be considered as an admission that such
24 dependent claims are not patentable over the cited references. Similarly, applicants’ decision not to discuss
25 differences between the prior art and every claim element, or every comment made by the Examiner, should
26 not be considered as an admission that applicants concur with the Examiner’s interpretation and assertions
27 regarding those claims. Indeed, applicants believe that all of the dependent claims patentably distinguish over
28 the references cited. However, a specific traverse of the rejection of each dependent claim is not required,
29 since dependent claims are patentable for at least the same reasons as the independent claims from which the
30 dependent claims ultimately depend.

1 Discussion of Doyle:

2 Doyle teaches a system that enables a user of a browser program to access and execute program
3 objects embedded in hypermedia documents. Program objects are embedded in hypermedia documents
4 with the <EMBED> tag, which employs the “type” attribute to indicate the type of application that the client
5 should execute. The <EMBED> tag points to the location of data for the application on which the client
6 operates. Doyle teaches that the application program accesses data on a remote server and takes advantage
7 of increased server processing capacity. Doyle further teaches that the type element of the <EMBED> tag
8 refers to a MIME extension. Because the application program and the data operated on by the application
9 program are separate, it is unlikely that all computers viewing a hypermedia document with such an
10 embedded program object will have the necessary application program installed to view the data. This
11 problem occurs because the data are not packaged *with* the code module necessary to present or operate on
12 that data, so a client that does not have the required code module or program installed cannot access the data
13 to present the data to a user.

14 Doyle does not teach or suggest a service container comprising *a code module, a data module, and*
15 *an ID*. In fact, Doyle explicitly teaches that the application program *is directed to the location of the data*,
16 typically on one or more remote servers, to take advantage of distributed computing techniques. In contrast
17 to Doyle, Claim 21 recites that the data and the code necessary to operate on that data are included in the
18 same service container. Clearly, it would not be obvious to include the code and the data in a service
19 container, as recited in this claim, in view of the teaching of Doyle.

20 Discussion of Held:

21 Held teaches a method and system for transparently executing code using a surrogate process, this
22 method being known in the art as using a Distributed Component Object Module or “DCOM.” Further,
23 Held discloses a method of finding, loading, and executing code, but not of representing data. The code can
24 be located in dynamic link libraries (DLLs), or in executable modules, and is loaded at the request of a client
25 program. Held teaches the use of interfaces, CLSID’s (i.e., globally unique class identifiers used in COM),
26 class factories, server DLLs, and many other techniques well known in the art. While Held teaches the
27 concept of a software interface, Held does not teach or suggest a service container that includes a data
28 object, a loader identification, and a code object enabled to present to the user data stored in the data object.
29 In particular, Held does not teach a data object and does not teach a standard, universal interface enabled to
30 allow any computer application that implements the interface to request a service container and execute that

1 service container's code to present that service container's data object to a user. The surrogate process of
2 Held is not equivalent to applicants' recited service container, because there is no disclosure that the
3 surrogate process includes a data object, a loader identification, and a code object that is able to present user
4 data stored in the data object.

5 Discussion of the Rejection of Independent Claim 21

6 The Examiner asserts that a "data object" as recited in Claim 21 is equivalent to a class factory, a
7 new object, or an existing object. While all objects could be loosely described as "data," because all objects
8 are stored in computer memory, one of ordinary skill in the art would understand that the term "data object"
9 means an object that stores particular information. However, applicants note that one of ordinary skill
10 would not understand from the cited art alone, or in combination, that a service container can include a data
11 object, a code object, and a loader identification that enable definition of a type of content unknown to a
12 computer program. There is really no reason to assert that a surrogate process includes each of these
13 elements.

14 Also, Claim 21 specifically recites "executing the at least one segment of computer code along with
15 the computer program to process the input and obtain the output comprising the type of content that is
16 unknown to the computer program." The Examiner has indicated that Doyle's disclosure of a list of
17 applications from which an application can be selected is equivalent to this recitation. However, this claim
18 further recites that "the plurality of segments of computer code and the at least one segment of computer
19 code *are not executable as an independent computer program,*" which is clearly a requirement that is not
20 met by a selected stand-alone application like that disclosed by Doyle. Held's disclosure of a second
21 program to perform a desired task on data is also not equivalent to applicants' recitation, since the second
22 program of Held is apparently able to be executed as an independent computer program. Applicants'
23 recitation of "at least one segment of computer code" that is executed with "the computer program to obtain
24 the output is not taught or suggested by the cited references, since each of these references uses a separate
25 application program that is executable as an independent program to accomplish its function, and does not
26 provide at least one segment of computer code that is executed along with the computer program to
27 accomplish the recited function. Accordingly, Claim 21 is patentable over Doyle and Held.

28 Since dependent claims are patentable for at least the same reasons as the independent claims on
29 which they depend, Claims 22-29 and 41-43 are patentable over the cited art for at least the same reasons.
30 The rejection of Claims 21-29 and 41-43 should be withdrawn.

1 Independent Claims 30 and 36 are respectively a system claim and a memory media claim. The
2 Examiner has rejected these claims over Doyle and Held for substantially the same reasons as indicated in
3 his rejection of Claim 21. Accordingly, for substantially the same reasons as set forth to traverse the
4 rejection of Claim 21 in view of these references, each of Claims 30 and 36 also patentably distinguish over
5 the cited combination of references. Further, dependent Claims 31-35, and 44-46, which ultimately depend
6 from Claim 30, and dependent Claims 37-40, and 47-49, which ultimately depend from Claim 36 are
7 patentable for at least the same reasons as Claims 30 and 36, respectively. Therefore, the rejection of
8 Claims 30-35, 44-46, Claims 36-40, and 47-49 should be withdrawn.

9 Claim 50 is an independent claim directed to a method for accessing multiple types of electronic
10 content from a program. However, Claim 50 is much more specific in reciting the details for carrying out
11 the method than method Claim 21. Yet, the Examiner simply applied the Doyle and Held references in
12 rejecting Claim 50 in substantially the same way as in rejecting Claim 21 and each of the other independent
13 claims. There is clearly little relationship between the disclosure of Doyle and Held and the detailed
14 recitation of Claim 50. For example, applicants recite “defining a plurality of service containers accessible
15 to the program, each service container corresponding to a specific function,” that each service contains a
16 data object, and that “each data object includes data required to enable the specific function corresponding to
17 the service container to be achieved.” There is no teaching or suggestion in Held of a surrogate process
18 including a data object that includes data required to carry out a specific function corresponding to the
19 surrogate process. Also, neither Doyle or Held teach or suggest that any equivalent of applicants’ recited
20 service containers include a “code object referencing at least one segment of programming code stored
21 separately from the service container, such that the at least one segment of programming code referenced by
22 the code object includes programming code required to enable the specific function corresponding to the
23 service container to be achieved.” Doyle and Held refer to separate application programs – not to segments
24 of programming code referenced by a code object. The cited art fails to teach or suggest a “loader
25 identification providing the program with information required to load the segment of programming code
26 referenced by the code object,” or any equivalent thereof.

27 Of even greater note, there is no discussion or suggestion of any of the following steps that are also
28 recited in Claim 50:

29 identifying one of the plurality of service containers whose corresponding specific
30 function is able to process the input;
parsing the service container thus identified, to determine:

1 data required to enable the specific function corresponding to the service
2 container to be achieved;
3 an identity of the at least one segment of programming code required to
4 enable the specific function corresponding to the service container to be achieved; and
5 information required to load each segment of programming code referenced
6 by the code object;
7 retrieving the at least one segment of programming code required to enable the
8 specific function corresponding to the service container to be achieved; and
9 executing the at least one segment of computer code under the control of the
10 computer program to process the input and obtain the output, wherein said computer
11 program is not directed to the at least one segment of computer code by any embedded link
12 referenced by said computer program.

13 The Examiner is respectfully asked to provide an indication of where either Doyle or Held teaches
14 or suggests “parsing the service container” (or parsing a surrogate process) to determine any information, let
15 alone applicants’ specific recitation of “data required to enable the specific function corresponding to the
16 service container to be achieved,” or “an identity of the at least one segment of programming code required
17 to enable the specific function corresponding to the service container to be achieved,” or “information
18 required to load each segment of programming code referenced by the code object.” Clearly, the Examiner
19 has ignored these recited details in Claim 50, since no attempt has been made to show where the art cited
20 teaches or suggests them. It is apparent that this claim patentably distinguishes over the references cited and
21 should also be allowed.

22 In view of the amendments and Remarks set forth above, it will be apparent that the claims in this
23 application define a novel and non-obvious invention, and that the application is in condition for allowance
24 and should be passed to issue without further delay. Should any further questions remain, the Examiner is
25 invited to telephone applicants’ attorney at the number listed below.

26 Respectfully submitted,

27 /ron anderson/
28 Ronald M. Anderson
29 Registration No. 28,829

30 RMA/DWF:elm